

Recommended Spectre Monte Carlo Modeling Methodology

Don O'Riordan

Cadence Design Systems

Version 1, 17 December 2003 This document describes the Spectre Monte Carlo modeling methodology. It does so by dissecting a example netlist. It also describes how to pick correlation coefficient and mismatch spread values for a particular process.

Last updated on May 11, 2006. You can find the most recent version at www.designers-guide.org. Contact the author via e-mail at riordan@cadence.com.

Permission to make copies, either paper or electronic, of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage and that the copies are complete and unmodified. To distribute otherwise, to publish, to post on servers, or to distribute to lists, requires prior written permission.

1 Example

The Monte Carlo Modeling Methodology of the Spectre Circuit Simulator is explained by way of a simple example. The example is a circuit that consists of a pair of resistors whose values exhibit random variations. The example netlist is given in Listing 1 in its entirety, and then the netlist is dissected and explained.

The first section of the file, *Process section*, contains process-specific information to describe the manufacturing process parameters, their statistical variations, and a model for the resistor that includes the process parameters in the calculation of the effective resistance. This section of the file may be created by manufacturing process experts, and placed in a separate file that gets included into the final Spectre circuit simulator netlist using an include statement.

The second section of the file, *Design-specific section*, contains design-specific information. For the example netlist, this section includes the circuit connectivity (two resistors, and corresponding current sources that feed them), the Monte Carlo analysis that the designer wants to run, and some further statistics, indicating a correlation between two devices in the circuit, namely the two resistors *R1* and *R2* in this particular case. This information is typically created by the circuit designer, either by manual additions to the netlist, or by running the Artist Monte Carlo tool that creates this from information entered using the Graphical User Interface.

The sections are now described in more detail.

2 Process Parameters

Process parameters are defined at the top of the file, as follows:

```
// Define resistor process parameters, including mismatch effects
parameters RSHSP=200      // sheet resistance
+           SPDW=0         // etching variation from ideal
+           XRSP=1          // device mismatch parameter
+           XRSPafac=100m   // RS mismatch parameter
+           XRSPbfac=1m     // RS mismatch parameter
+           RSHSPnom=200    // sheet resistance nominal value
```

Manufacturing process parameter values are given for sheet resistance ($RSHSP = 200$ ohms/square) and etching variation over/under etch ($SPDW$). The nominal value for sheet resistance is 200 ohms per square. This will later be modified to take on some different values during a Monte Carlo analysis.

The nominal value for $SPDW$ is 0. This will later be modified to take on some non-zero values during a Monte Carlo analysis.

A resistor device mismatch parameter ($XRSP$) is also declared, whose nominal value is 1. This parameter will also be subject to statistical variation during a Monte Carlo analysis involving mismatch statistics.

Two factors, *A* and *B*, are then declared for the statistical matching of resistance parameter $XRSP$. These are declared as $XRSPafac=100m$ and $XRSPbfac=1m$ respectively. These parameters are constants, and their meaning is discussed in Section 3.

LISTING 1 *A simple example that demonstrates Spectre's Monte Carlo modeling methodology.*

```

// A simple resistor circuit where the resistors exhibit random variations
simulator lang=spectre

// Process section
// Define resistor process parameters, including mismatch effects
parameters RSHSP=200 // sheet resistance
+           SPDW=0 // etching variation from ideal
+           XRSP=1 // device mismatch parameter
+           XRSPafac=100m // RS mismatch parameter
+           XRSPbfac=1m // RS mismatch parameter
+           RSHSPnom=200 // sheet resistance nominal value

// A resistor subckt model that includes geometrical and process effects
inline subckt RPLR (A B)
parameters Rnom=1 // input, nominal resistance value
+           WB=10u // input, resistor width
+           LB=Rnom*WB/RSHSPnom // length = R*W/RSH
+           AB=LB*(WB+SPDW) // effective area, with etching effect
RPLR (A B) resistor r=RSHSP*LB/(WB+SPDW)
+           *(1+(XRSPafac/sqrt(AB)+XRSPbfac)*(XRSP-1))
ends RPLR

// A statistics block that specifies process variations
statistics {
    process {
        vary RSHSP dist=gauss std=5 // apply variations in sheet resistance
        vary SPDW dist=gauss std=0.25 // apply variations in etching
    }
    mismatch {
        vary XRSP dist=gauss std=1 // apply resistor mismatch
    }
}

// Design-specific section
// Two resistors, 4k ohms nominal, different geometries
R1 (1 0) RPLR Rnom=4kOhm WB=5 // 5 units wide, self-calculates length
R2 (2 0) RPLR Rnom=4kOhm WB=10 // 10 units wide, self-calculates length

// Current source biasing
J1 (0 1) isource dc=1mA // force 1mA through R1
J2 (0 2) isource dc=1mA // force 1mA through R2

// Monte Carlo analysis specification
m1 montecarlo saveprocessparams=yes processscalarfile="../process_simple.dat"
+           numruns=3 variations=mismatch seed=10 {
    dcopy dc
    export v1=oceanEval("v(\\"1\\)") // measure voltage across R1
    export v2=oceanEval("v(\\"2\\)") // measure voltage across R2
    export v1_2=oceanEval("v(\\"1\\") - v(\\"2\\)") // and the voltage difference
}

// match pairs, specify correlation coefficients
statistics {
    correlate dev=[R1 R2] cc=0.75 // correlate the resistors
}

```

Finally, a parameter RSHSPnom is declared, whose value represents the nominal value of the resistor sheet resistance. This parameter will also be kept constant during all simulations, though its counterpart RSHSP will be subject to statistical variation.

The next section of the file declares an inline subcircuit with which to model the resistor device, and the effects of process and mismatch parameter variations upon that device.

3 Resistor Modeling

```
// A resistor subckt model that includes geometrical and process effects
inline subckt RPLR (A B)
parameters Rnom=1 // input, nominal resistance value
+           WB=10u // input, resistor width
+           LB=Rnom*WB/RSHSPnom // length = R*W/RSH
+           AB=LB*(WB+SPDW) // effective area, with etching effect
           RPLR (A B) resistor r=RSHSP*LB/(WB+SPDW)
+           *(1+(XRSPafac/sqrt(AB)+XRSPbfac)*(XRSP-1))
ends RPLR
```

The inline subckt RPLR has two ports (terminals) A and B respectively. A resistor primitive is connected across the two ports. The value of the resistor primitive is a function of the subcircuit parameters and the global process parameters. The subcircuit input parameters include the nominal value of the resistor (Rnom, defaults to 1), and the nominal width of the resistor (WB, defaults to 10u). The subcircuit parameters also include several parameters that hold intermediate values, including LB and AB.

The parameter LB represents the length of the resistor, and is calculated from the nominal value of the resistor, the width of the resistor, and the sheet resistance nominal value. It is possible to redefine this inline subckt so that the input parameters are WB and LB representing the resistor geometry, and have the nominal resistance value be calculated from these. However, it is more typical for a designer to want to specify the resistance value and width, and figure out the length at layout time.

$$LB = \frac{Rnom \times WB}{RSHSPnom} \quad (1)$$

Note that the nominal (constant) value of RSHSPnom is used in calculating the length, as this is the value that the designer will use at layout time to determine the resistor length that gives the desired resistance for a given width. Once a Monte Carlo simulation begins, the parameter RSHSP will be subject to statistical variation. If the parameter RSHSP were to be used in the expression for LB instead of the nominal value RSHSPnom, it would lead to different resistor length values being calculated.

The parameter AB contains the effective area of the resistor, taking over/under etching variations into account. It is assumed that these variations do not affect the length of the resistor, but do affect the width. Resistors tend to be much longer than they are wide, hence etching variations in length can be ignored. The expression for parameter AB contains a reference to the statistical process parameter SPDW, which has a nominal value of zero, but is varied statistically during a Monte Carlo analysis that includes process variations. This will have the effect of varying AB, which is subsequently used in the calculation of the effective resistance value.

The resistor value itself is calculated using the following large expression:

$$r = \frac{RSHSP \times LB}{(WB + SPDW)} \times \left(1 + \left(\frac{XRSPafac}{\sqrt{AB}} + XRSPbfac \right) \times (XRSP - 1) \right) \quad (2)$$

3.1 Nominal Resistance Value

Consider the process variation only case where statistical mismatch parameter XRSP is set to 1. The equation reduces to the following:

$$r = \frac{RSHSP \times LB}{WB + SPDW} \quad (3)$$

If we assume no process variations are applied (i.e. we are not performing a Monte Carlo analysis), the value of the resistance is RSHSP*LB/WB, assuming SPDW takes on its nominal value of zero.

Substituting in the equations for LB=(Rnom*WB/RSHSPnom), and assuming SPDW = 0, and RSHSP = RSHSPnom = 200, we get the following in the nominal case:

$$\begin{aligned} r &= \frac{RSHSP \times \left(Rnom \times \frac{WB}{RSHSPnom} \right)}{WB} \\ &= Rnom \end{aligned} \quad (4)$$

Hence, for the nominal run, $r = Rnom$ as intended.

3.2 Resistance Value including Process Variations

Here, we assume that no mismatch variation is performed, so that parameter XRSP is fixed at its default value of 1. The resistor value is calculated from (2), which reduces to (3).

If we substitute the value of LB, we get

$$r = \frac{RSHSP \times Rnom \times \frac{WB}{RSHSPnom}}{WB + SPDW} \quad (5)$$

When we perform a Monte Carlo analysis that includes process variations only, the value of RSHSP and SPDW are sampled differently for each Monte Carlo run. This is because these two process parameters have variations specified in the process section of the statistics block. For more information, see Section 4.

Hence, for each Monte Carlo run, the effective resistance r will be increased or decreased somewhat by the change in RSHSP and a non-zero value of the over/under etch factor SPDW. Equation 2 is re-evaluated for every Monte Carlo run. As this reduces to (5), the resistance r will vary about Rnom by some value, and for any given resistor, (for example, instances R1 and R2) the resistance values will be different for each run.

4 Process Variations and Geometrical Effects

The actual resistor instances in the netlist file are:

```
// Two resistors, 4k ohms nominal, different geometries
R1 (1 0) RPLR Rnom=4kOhm WB=5    // 5 units wide, self-calculates length
R2 (2 0) RPLR Rnom=4kOhm WB=10   // 10 units wide, self-calculates length
```

Resistor $R2$ is drawn using a width of 10 units, and thus takes up four times as much area as resistor $R1$, which only has a width of 5 units. (i.e. $R2$ needs to be twice as long as $R1$ so that both resistors end up with the same nominal resistance of 4k).

This also means that resistance $R2$ is less susceptible to the over/under etching variations than is $R1$. For a large value of WB , the value of $(WB + SPDW)$ is closer to the value of WB than for a small value of WB . Since the equation for process variations only reduces to (5), we can see that large area resistors are less susceptible to etching variations than small area resistors of equal resistance. The design trade-off becomes one of large chip area and manufactured circuit performance close to the nominal simulated result, versus one of small chip area and large circuit performance variations.

By modeling the resistor value using (2), we have successfully modeled this statement of reality, and incorporated geometrical effects in the process and simulation model.

5 Resistance Value Including Mismatch Variations

In this section, we assume that only mismatch variation is performed so that parameter $XRSP$ is no longer fixed at its default value of 1, but is free to vary from one resistor inline subckt instance to another. Mismatch-only variation means that the parameters $RSHSP$ and $SPDW$ are held constant at their nominal values of 200 ohms/square and zero respectively.

Again, the full resistor value is given by (2).

Assuming $RSHSP = RSHSPnom$ and $SPDW = 0$ reduces the equation to:

$$r = \frac{RSHPNom \times LB}{WB} \times \left(1 + \left(\frac{XRPafac}{\sqrt{AB}} + XRSPbfac \right) \times (XRSP - 1) \right) \quad (6)$$

If we again substitute the value of LB , we get

$$r = \frac{RSHPNom \times \frac{Rnom \times WB}{RSHSPnom}}{WB} \times \left(1 + \left(\frac{XRPafac}{\sqrt{AB}} + XRSPbfac \right) \times (XRSP - 1) \right) \quad (7)$$

This reduces to:

$$r = Rnom \times \left(1 + \left(\frac{XRPafac}{\sqrt{AB}} + XRSPbfac \right) \times (XRSP - 1) \right) \quad (8)$$

This can also be stated as follows:

$$r = Rnom \times (1 + \varepsilon) \quad (9)$$

where ε represents a small mismatch term and is

$$\varepsilon = 1 + \left(\frac{XRPafac}{\sqrt{AB}} + XRSPbfac \right) \times (XRSP - 1) \quad (10)$$

Since the A and B factors, XRSPafac and XRSPbfac, are constant, and the area AB is effectively constant if no process variations are considered (here we consider mismatch variation only), parameter XRSP is responsible for introducing variations in the individual resistor instance resistance values. A mismatch only Monte Carlo analysis using the Spectre circuit simulator results in a different value of XRSP for every subcircuit (resistor in this case) instance for any given Monte Carlo run. Hence, two otherwise identical resistor instances (*R1* and *R2* are close, but not quite identical due to different geometry: if both had a width of *W2*, they would be identical), would in fact take on slightly different resistor values, as each would get a different value of XRSP during the same Monte Carlo run.

This allows us to model the fact that in reality, no two resistor instances on the same die are identical. By placing the resistor instance in a subckt (an inline subckt in this example, but regular subckts will also suffice), we gain the ability to apply separate mismatch variations to each subckt (and hence resistor) instance, and model this reality in our simulation.

6 Mismatch Variations and Geometry Effects

For mismatch variation, we defined (9) and (10) where the effective device area is

$$AB = LB \times (WB + SPDW) \quad (11)$$

Here we see that the value for resistance consists of the nominal value *Rnom*, plus *Rnom* multiplied by some small mismatch term. The mismatch term is a function of the device area (inversely proportional to square root of device area, AB), and if we temporarily assume that XRSP is close to 1, then the value of the mismatch term is close to zero. For any given fixed value of XRSP, large geometry devices have a smaller mismatch term than small geometry devices, because of the $1/\sqrt{AB}$ term in the equation.

The equation relating to the inverse square root of the area above is based on the well-published Pelgrom matching model, details of which are available in the literature and are beyond the scope of this document. It is expected that process experts will be able to furnish the values of XRSPafac and XRSPbfac for any given manufacturing process based on test chip results.

The resistance equation given here is relatively simple. Process experts may create inline subcircuits containing more complex equations and interactions representing their understanding of their process.

The concepts presented here may be extended to other device types, such as MOSFETS and BJT transistors.

Here, we have modeled the fact that large geometry devices tend to exhibit better matching characteristics than small geometry devices.

7 Matched Devices and *Correlate* Statements

Analog designers are happy to leverage the fact that devices placed adjacent to one another on a particular chip tend to exhibit better matching properties than devices that

are separated by a large distance. The Spectre circuit simulator will model and simulate this happy fact using device correlation statements, as contained in the example:

```
// match pairs, specify correlation coefficients
statistics {
    correlate dev=[R1 R2] cc=0.75      // correlate the resistors
}
```

This statement states that all mismatch parameters for instances *R1* and *R2* will match each other with a high correlation coefficient of 0.75 (a value of 1.0 means perfect correlation, or ideal matching).

In this particular example, the only mismatch parameter specified is XRSP. The example also assumed two resistor instances *R1* and *R2*.

Let us temporarily consider a slightly different circuit, which assumes more than two resistors, all with identical geometries:

```
// four resistors, 4k ohms nominal, identical geometries
R1 (1 0) RPLR Rnom=4k WB=10
R2 (2 0) RPLR Rnom=4k WB=10
R3 (3 0) RPLR Rnom=4k WB=10
R4 (4 0) RPLR Rnom=4k WB=10
```

From one simulator perspective, all four resistors are identical. However, from the perspective of a Monte Carlo simulation including mismatch, the resistors are different from each other. The device correlation statement:

```
correlate dev=[R1 R2] cc=0.75      // correlate the resistors
```

states that the group of resistors [*R1 R2*] are less different from each other than are all other resistor pairs. It means that *R1* and *R2* are somehow more closely matched. However, *R1* and *R3*, *R1* and *R4*, *R2* and *R3*, *R2* and *R4* and *R3*, and *R4* are all significantly different from each other when considered pair-wise. Only the pair *R1* and *R2* is special, as indicated by the presence of the non-zero correlation coefficient. All other pairs are considered to have a zero correlation coefficient, indicating that they are not matched. The circuit designer would need to layout *R1* and *R2* very close to each other, and interleaved if possible, in order to have this also happen in reality. It turns out that this is what designers actually do in practice when they need to have close matching components in differential circuit applications such as operational amplifiers for example.

The device correlation statement allows the designer to indicate to the simulator that groups of devices will be matched up in the layout. The simulator in turn will ensure that the random values generated for the mismatch parameters for correlated devices are also correlated during a Monte Carlo mismatch simulation. Thus, the value of mismatch XRSP for instance *R1* will be pretty close to the value of XRSP for instance *R2* for any given Monte Carlo run, but these may be much different than the values of XRSP for instances *R3* and *R4*, which are assumed to be placed at some distance from each other, and at some distance from instances *R1* and *R2*.

We now address the question of how close is close.

8 Choosing Correlation Coefficients and Mismatch Factors

Choosing a correlation coefficient must be performed somewhat in concert with extracting the values of the mismatch parameters XRSPafac and XRSPbfac. A guideline is given below:

The spread of ε in (9) is a function of device geometry (inversely proportional to square root of area). However, this spread is also proportional to the A and B factors XRSPafac and XRSPbfac respectively. The mismatch term is calculated by (10).

For a given value of XRSP, and a constant area AB, the size of the mismatch term is directly proportional to XRSPafac. Hence, the spread of the mismatch term is also directly proportional to XRSPafac.

If we now consider the correlation coefficient again, the spread of correlated device parameters will be smaller than the spread of non-correlated device parameters. This reduction in spread can be obtained from $\sqrt{1 - cc}$ where cc is the correlation coefficient of the correlated devices.

If we assume that $cc = 0.75$, the reduction in spread for correlated devices is $\sqrt{1 - 0.75} = \sqrt{0.25} = 0.5 = 1/2$

So devices that are specified with a correlation coefficient of 0.75 have a spread that is half the spread exhibited by non-correlated devices.

If we reverse this sentence, we can state that non-correlated devices have a spread that is twice that of correlated devices (assuming $cc = 0.75$).

This statement tells us that if we can measure the spread of matched (correlated) devices, and we assume a matching correlation coefficient of 0.75, we can predict the spread of non-matched (i.e. non-correlated) devices by multiplying by 2.

Specifically, this means that if we measure XRSPafac for matched resistors, the value of XRSPafac for non-matched resistors will be twice the measured value, assuming $cc=0.75$.

Hence, if we measure XRSPafac=50m for matched resistors from a test chip, we can use XRSPafac=100m in a process file to properly scale the spread of mismatch parameters for non-matched devices if we assume that a cc of 0.75 will be used during simulation for matched devices.

Reading this backwards, we now assume that the value of XRSPafac in the process file represents the value of the non-matched resistors (XRSPafac=100m). We can reduce the spread for matched resistors by a factor of 2 (back to XRSPafac=50m, as measured) by specifying $cc = 0.75$ for those matched resistors!

So by choosing XRSPafac as twice the measured value for matched devices, and choosing a corresponding correlation coefficient of 0.75 for matched devices, we can model a third level of statistical variation implicitly.

8.1 Another Example

If instead we chose the value of XRSPafac for non-matched devices as three times the measured value of this parameter for closely-matched devices, then we would need to choose a different correlation coefficient as follows:

$$1/3 = \sqrt{1 - cc} = 0.333$$

$$(1 - cc) = (0.333) * (0.333)$$

$$= (0.111)$$

$$cc = (1 - 0.111) = 0.889$$

The ratio of XRSPafac between matched and non-matched devices is a function of the manufacturing process. Values of 0.5 are fairly typical. The corresponding value of $cc = 0.75$ is thus correspondingly fairly typical, which is why we use this value to seed the entry in the ADE (*a.k.a.* Artist) Monte Carlo Graphical User Interface.

However, process experts are completely free to decide upon (measure) a ratio for their own process, and communicate the corresponding correlation coefficient for such matched devices to their design community.

To conclude, the Spectre modeling approach assumes that:

1. Mismatch A scale factors (such as parameter XRSPafac in the example above) have some known ratio for matched versus non-matched devices on the same chip
2. Mismatch A scale factors (such as parameter XRSPafac in the example above) specified in the process files and included into the final Spectre circuit simulator netlist are assumed to be the scale factors for non-matched devices, introducing an implicit third degree of spread in non-matched devices
3. Matched devices are indicated by specifying a correlation coefficient that represents the known ratio. The simulator will perform the correlation in its selection of random values, thus reducing the spread of matched devices by the known factor, completely analogous to how a designer will physically place devices close together on chip in order to reduce their spread.

The advantages of this technique include less random number generation (since the number of matched pairs/devices in a chip is typically much less than the number of non-matched devices) and a modeling and simulation approach that is closer to what the designers actually do in real life.

8.2 Even Closer Matching

If XRSPafac is specified assuming a scale factor of two ($1/2 = 0.5$), then the corresponding correlation coefficient is 0.75.

If, however, a layout engineer interlaces the resistors to make them very well matched (or laser trimming or some other technique), the actual spread of XRSPafac for these well-matched devices may be actually smaller than the assumed value of $\text{XRSPafac}/2$!

This can be modelled by choosing an even larger correlation coefficient cc . In fact, if σ_R is the real actual spread of tightly matched devices, and σ_A is the assumed spread specified in the process parameter file, the actual spread can be simulated by choosing a larger correlation coefficient as follows:

$$cc2 = 1 - \left(\frac{\sigma_R}{\sigma_A} \right)^2 \quad (12)$$

Again, a designer is free to use $cc2$ on devices instances that he knows will be very well matched in layout.

9 Process Parameter Correlation

Correlated process parameters are specified using the process parameter correlation statement. The following example illustrates how this may be used:

```
// Process and mismatch parameters.
parameters RSHSP=200           // sheet resistance
+           RSHPI=5k           // pinched sheet resistance
+           SPDW=0 SNDW=0      // etching variation from ideal
+           XRSP=1             // device mismatch parameter
+           XRSPafac=100m      // RS mismatch parameter
+           XRSPbfac=1m        // RS mismatch parameter
+           RSHSPnom=200       // sheet resistance nominal value
+           RSHPInom=5k        // sheet resistance nominal value
+           FRSHPI=RSHPI/RSHPInom // ratio of PI sheet res to nominal
```

Here RSHSP and RSHPI are different process parameters, representing two different sheet resistances. For a given manufacturing process, sheet resistance parameters are likely to be correlated. In order to model this effect, the process parameter correlation statement is used within the Spectre circuit simulator statistics block:

```
// A statistics block, which specifies process variations in two sheet resistance
// parameters and over/under etch values, and also varies a resistor mismatch
// parameter. In addition, a correlation statement is included that specifies
// the correlation between the two sheet resistance parameters.
statistics {
  process {
    vary RSHSP dist=gauss std=5           // apply variations in sheet res
    vary SPDW dist=gauss std=0.25        // apply variations in etching
  }
  correlate param=[RSHSP RSHPI] cc=0.6 // correlate sheet resistance!
  mismatch {
    vary XRSP dist=gauss std=1           // apply resistor mismatch
  }
}
```

In this example, the following statement was used to indicate that the process parameters RSHSP and RSHPI are reasonably correlated:

```
correlate param=[RSHSP RSHPI] cc=0.6 // correlate sheet resistance!
```

It is important to note that this correlation is completely different from the mismatch factors and correlation coefficients discussed in previous sections. In this example, we are back to the process parameter variations only scenario. Here, we are stating that for any given Monte Carlo run (which includes process variations), the random values chosen for parameters RSHSP and RSHPI will be correlated, and not drawn independently from each other. This is true even if no mismatch variations are specified.

For example:

Process parameter RSHSP is used in the calculation of resistance values of diffused resistors as in the example above.

The parameter RSHPI can be used in the calculation of the saturation current and current gain parameters (IS and BF) of a bipolar transistor manufactured on the same process.

parameters

```

+      IS=1e-18*FRSHPI*AE*(1+(XISNafac/sqrt(AE)+XISNbfac)
+      *(dIS/2+XISN-1)/sqrt(MULT))
+      BF=100*FRSHPI*(1+(XBFNafac/sqrt(AE)+XBFNbfac)
+      *(dBF/2+XBFN-1)/sqrt(MULT))

```

Here, both IS and BF depend on FRSHPI, which is calculated from the following process parameter declaration (see process and mismatch parameters declared above in this example)

```
FRSHPI=RSHP/RSHPInom // ratio of PI sheet res to nominal
```

FRSHPI=RSHP/RSHPInom is a function of RSHP, which varies in tandem with RSHP.

Since both transistors and resistors are manufactured on the same process, it is likely that these parameters will be somewhat correlated in reality. It is thus important to model these correlations as shown above using the process parameter correlation statement.

For a given Monte Carlo run, the transistors will be somewhat correlated to the resistors, matching reality.

10 If You Have Questions

If you have questions about what you have just read, feel free to post them on the *Forum* section of *The Designer's Guide Community* website. Do so by going to www.designers-guide.org/Forum.